Basic Optimization Algorithms

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics

- Importance of Optimization in machine learning
- How learning differs from optimization
- Challenges in neural network optimization
- Basic Optimization Algorithms
  1. Stochastic Gradient Descent
  2. Momentum
  3. Nesterov Momentum
- Parameter initialization strategies
- Algorithms with adaptive learning rates
- Approximate second-order methods
- Optimization strategies and meta-algorithms

Stochastic Gradient Descent

- We have seen:
  - Gradient descent that follows the gradient of an entire training set downhill
  - This can be accelerated considerably by SGD which follows the gradient of randomly selected minibatches downhill
- SGD and its variants are the most used optimization algorithms for ML in general and deep learning in particular
  - Average gradient on a minibatch is an estimate of the gradient

Following gradient estimate downhill

- Algorithm: SGD update at training iteration k

Require: Learning rate ϵ_k.
Require: Initial parameter $\boldsymbol{\theta}$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow +\frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \hat{\boldsymbol{g}}$
end while

- A crucial parameter for Algorithm SGD is the learning rate ε
  - It is necessary to let the learning rate decrease
    - At iteration k it is ε_k

Need for decreasing learning rate

- Batch gradient descent can use a fixed learning rate
  - Since true gradient becomes small and then 0
- SGD has a source of error
  - Random sampling of m training samples
  - Sufficient condition for SGD convergence $\boxed{\sum_{k=1}^{\infty} \varepsilon_k = \infty \qquad \sum_{k=1}^{\infty} \varepsilon_k^2 < \infty}$
  - Common to decay learning rate linearly until iteration τ: $\varepsilon_k=(1-\alpha)\varepsilon_0+\alpha\varepsilon_\tau$ with $\alpha=k/\tau$.
  - After iteration τ, it is common to leave ε constant

2. The Momentum method

- SGD is a popular optimization strategy
- But it can be slow
- Momentum method accelerates learning, when:
  - Facing high curvature
  - Small but consistent gradients
  - Noisy gradients
- Algorithm accumulates moving average of past gradients and move in that direction, while exponentially decaying

Momentum definition

- Introduce variable $\boldsymbol{v}$, or velocity
- It is the direction and speed at which parameters move through parameter space
- Momentum is physics is mass times velocity
- The momentum algorithm assumes unit mass
- A hyperparameter α ε $[0,1)$ determines exponential decay

Momentum update rule

- The update rule is given by

$$\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \varepsilon \nabla_\theta \left(\frac{1}{m} \sum_{i=1}^{m} L\left(f(x^{(i)}; \theta), y^{(i)} \right) \right)$$

$$\theta \leftarrow \theta + \boldsymbol{v}$$

- The velocity $\boldsymbol{v}$ accumulates the gradient elements $\nabla_\theta \left(\frac{1}{m} \sum_{i=1}^{m} L\left(f(x^{(i)}; \theta), y^{(i)} \right) \right)$
- The larger α is relative to ε, the more previous gradients affect the current direction
- The SGD algorithm with momentum is next

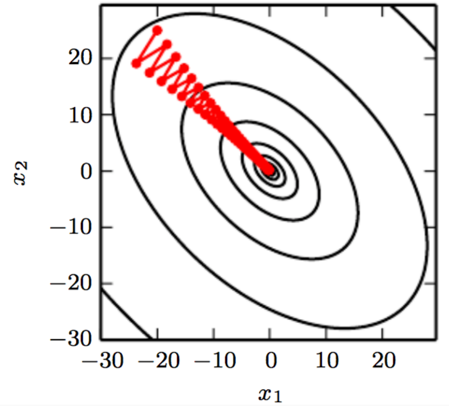
SGD algorithm with momentum

- Stochastic gradient descent with momentum

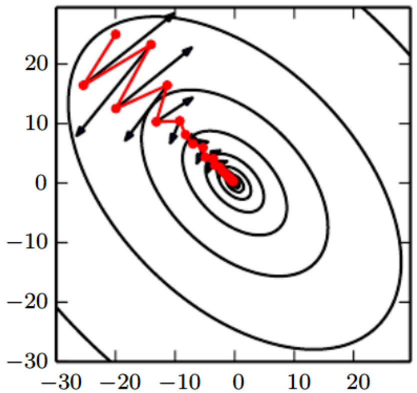
Require: Learning rate ϵ, momentum parameter α.
Require: Initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$.
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient estimate: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
Compute velocity update: $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$
end while

Momentum

- SGD without momentum



- SGD with momentum



Nesterov Momentum

- A variant of the momentum algorithm
- An accelerated gradient method
- It applies a correction factor to the standard method

Require: Learning rate ϵ, momentum parameter α.
Require: Initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$.
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ with corresponding labels $\boldsymbol{y}^{(i)}$.
    Apply interim update: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta} + \alpha \boldsymbol{v}$
    Compute gradient (at interim point): $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\tilde{\boldsymbol{\theta}}} \sum_i L(f(\boldsymbol{x}^{(i)}; \tilde{\boldsymbol{\theta}}), \boldsymbol{y}^{(i)})$
    Compute velocity update: $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$
  end while